

УДК 004.896

С.В. Титенко

## ГЕНЕРАЦІЯ ТЕСТОВИХ ЗАВДАНЬ У СИСТЕМІ ДИСТАНЦІЙНОГО НАВЧАННЯ НА ОСНОВІ МОДЕЛІ ФОРМАЛІЗАЦІЇ ДИДАКТИЧНОГО ТЕКСТУ

### Вступ

Українське суспільство все гостріше відчуває потребу в розповсюдженні навчальних процесів у найрізноманітніших сферах життя. Останнім часом освітні процеси, виходячи за межі освітніх закладів, дуже активно впроваджуються в такі сфери, як бізнес, корпоративна освіта, підвищення кваліфікації працівників різних галузей.

Найбільш зручною формою навчання тут є дистанційна. Ключовими складовими дистанційної освіти є контроль і діагностика знань, реалізація яких в умовах відсутності безпосереднього контакту між учасниками навчального процесу викликає певні технічні і організаційні труднощі. Однією з найбільш поширених дистанційних форм перевірки знань є комп'ютерне тестування. Науковці і спеціалісти в галузі електронної освіти і інформаційних технологій зробили значний внесок у розробку різних методів реалізації тестування. У праці П. Брусиловського [1] подано детальний огляд життєвого циклу тестових завдань в освітній Web-системі, що включає в себе такі стадії, як підготовка, подання і оцінювання. Як правило, стадія підготовки тестових завдань залишається слабко автоматизованою і вимагає безпосередньої роботи викладача.

У той час, як багато досліджень у галузі комп'ютерного контролю знань зосереджені на питаннях валідності і надійності тестів [2], питання формування самого банку завдань у більшості випадків залишається виключно прерогативою викладача, який працює без використання інтелектуальних засобів автоматизації даного процесу. Дійсно, намагання автоматизувати формування тестових завдань наштовхують на сферу штучного інтелекту і проблеми формалізації знань та їх подальшого використання в генерації тестів.

Традиційний підхід до створення засобів тестування фактично являє собою комп'ютери-

зацію ручного тестування. Суть такого підходу полягає у використанні інформаційно-комунікаційних технологій замість роботи на папері, що дає додаткові можливості щодо управління формуванням тестів з банку створених завдань та автоматичної перевірки результатів. Одночасно і перевагою, і недоліком даного підходу є ручне, неавтоматичне створення завдань. Завдання, професійно розроблені експертом, мають високу якість і зрозумілість. У випадку конкретних статичних тестів з'являються широкі можливості для зосередження на питаннях валідності і надійності. Однак істотним недоліком підходу є висока трудомісткість самого процесу з формування тестових завдань. Ця проблема підсилюється задачею захисту від недобросовісного проходження тесту, що передбачає наявність великого банку завдань і динамічну композицію на його основі індивідуального тесту для попередження "списування". Крім того, викладач, що створює тестові завдання, несе педагогічну відповідальність за адекватність змісту контролю навчальному контенту. Тому це вимагає від розробника тестів не тільки експертних знань з предметної області, але й знання і глибокого аналізу безпосереднього навчального матеріалу, який подається студентам дистанційної форми навчання як основне джерело для оволодіння знаннями.

### Постановка задачі

Метою статті є аналіз існуючих методів автоматизації побудови тестових завдань і розробка моделі, що дала б можливість подолати виявлені в ході аналізу недоліки і організувати автоматизоване тестування із врахуванням таких вимог:

- простота створення бази знань, що слугуватиме основою для генерації тестових завдань;
- можливість побудови тестових завдань різних типів;
  - лексична зрозумілість тестових завдань;
  - можливість гнучкої побудови тестів залежно від необхідної ділянки навчального контенту, мети і стратегії тестування;
- можливість використання результатів тестування для адаптивного налаштування індивідуального навчального процесу;
- можливість інтеграції моделі з ієрархічно-мережевою моделлю освітнього контенту Tree-Net [3–5].

### Методи автоматизації побудови тестових завдань

З метою інтенсифікації і спрощення процесів підготовки дистанційних навчальних курсів з тестовими блоками науковцями було запропоновано різні підходи до автоматизації створення тестових завдань. Одним із перспективних і порівняно нескладних у реалізації є підхід параметризованих тестів [6]. Суть підходу полягає в поданні різним студентам шаблонного завдання, яке відрізнятиметься певними параметрами, що генеруються автоматично. Відповідь вводиться з клавіатури. Таким чином, кожен студент отримує індивідуальне завдання, а система за певною формулою чи алгоритмом, підставляючи параметри, отримує правильну відповідь для подальшої перевірки відповіді, введеної студентом. Недоліком підходу є його вузька предметна спрямованість. Так, параметризовані тести добре підходять для організації контролю практичних навичок у точних науках і програмуванні [7], проте не можуть використовуватись для перевірки теоретичних знань, а також контролю в гуманітарних науках.

Певного поширення в дослідженнях автоматизації контролю знань отримав підхід застосування семантичних мереж для автоматизації побудови тестів [8, 9]. Наріжним каменем семантичних мереж є так звані тріади [8–10]: сутність 1—відношення—сутність 2. Наприклад, є такі сутності: “процедура” і “програма”. В такому випадку між ними можна встановити відношення типу “є частиною”. Тоді отримуємо: сутність “процедура” “є частиною” сутності “програма”. Завдання тесту будується шляхом опущення одної з ланок тріади і постановкою запитання про відсутню ланку. Перевагою даного підходу є здатність системи міркувати знаннями з предметної області. Недолік полягає у великих витратах при складанні завершеної цілісної семантичної мережі, яка б коректно відображала дану предметну область, що вивчається. Ще одним недоліком підходу є лінгвістична незрозумілість і, часом, недоцільність завдань, що генеруються. Так, на основі семантичної мережі часто ставляться запитання про такі особливості об’єктів предметної області, що не мають у даному навчальному контексті педагогічної цінності. Подібні недоліки виникають на основі проблеми, характерної для класичних моделей знань штучного інтелекту, яку можна назвати проблемою всеосвіченості. Так, вимогою семантичних мереж є

повна формалізація предметної області, тоді як вивчення матеріалу відбувається в певному контексті, тому при генерації завдань у тест часто попадають запитання, відповідь на які не свідчить про досягнення або недоліки в навчанні студента. Вимога всеосвіченості, або всеосяжної формалізації, потребує великих зусиль для побудови адекватної моделі, яка при цьому часто не забезпечує належної педагогічної доцільності в процесі її використання для освітніх цілей і тестування.

Ці та інші проблеми властиві для випадків, коли робиться змога застосувати класичні моделі штучного інтелекту (ШІ) для освітніх завдань. Незважаючи на те, що як в моделях ШІ, так і в навчальних системах об’єктом моделювання є знання, та все ж принципова різниця тут полягає в тому, що в першому випадку метою моделювання є надання системі здатності “знати” певну предметну область, щоб розв’язувати деякі прикладні задачі, натомість у другому випадку метою моделювання є надання системі здатності “навчати” людину знанням певної предметної області [11]. Така принципова різниця в постановці задач вимагає розробки спеціальних моделей формалізації знань для освіти і організації на їх основі автоматизованого тестування. Основним засобом передачі навчальної інформації студенту в дистанційному навчанні є текст — саме в ньому в зрозумілій для навчання формі подані знання. У зв’язку з цим вважаємо за доцільне покласти в основу моделі саме освітній текст, зосередившись на методах його формалізації з метою автоматизації побудови тестових завдань.

### Понятійно-тезисна модель формалізації дидактичного тексту

Понятійно-тезисна модель (ПТМ) формалізації дидактичного тексту розробляється на стику багатьох наукових галузей, серед яких: інженерія знань як напрямок штучного інтелекту; педагогіка, а саме її розділ — дидактика, що розкриває правила викладання; інтерпретація (герменевтика, екзегетика), що вивчає правила тлумачення текстів [12, 13]; лінгвістика і її розділ семантика, що вивчають закономірності природної мови і проблеми, пов’язані із змістом, значенням і інтерпретацією лексичних одиниць; технології розробки Web-систем, що є інструментом реалізації.

В освітньому процесі можна виділити такі чотири складові: передача навчальної інфор-

мації, соціальна складова, яка включає в себе спілкування між учасниками освітнього процесу, практичні заняття і контроль знань. Найперша задача дистанційної освіти – спрощення і оптимізація передачі навчальної інформації від носія знань до студента. Історично склалися, що в традиційних формах навчання початковою формою передачі інформації є вербальний спосіб: наприклад, викладач читає лекцію. Натомість у дистанційному, віддаленому від викладача, навчанні основною формою передачі навчальної інформації стає текст, а точніше, гіпертекст і мультимедіа-вміст, тобто основним транспортним засобом, або ж “протоколом”, передачі знань є текст, в якому міститься навчальна інформація. Це і є головною підвалиною понятійно-тезисної моделі знань.

Наріжним каменем структури ПТМ є така сутність, як поняття, предмет обговорення, деякий об’єкт із предметної області, про який у навчальному матеріалі є відомості. Поняття вказує на деякий об’єкт із предметної області, про який іде мова і який подається для вивчення студенту. Наприклад, в курсі “Алгоритмічні мови програмування” можна виділити такі поняття, як “процедура”, “цикл”, “програма”, “змінна”, “життєвий цикл програми” тощо. Для курсу “Програмування в середовищі Delphi” можна було б виділити поняття “об’єкт”, “подія”, “клас”, “форма”, “компонент TEdit” тощо. Множина понять у системі позначається так:

$$C = \{c_1, \dots, c_{n_1}\}.$$

Для подання знань про поняття ПТМ містить спеціальні структурні елементи – тези. Теза – це відомість про поняття. Тезу можна порівняти з ознакою, характеристикою або ж з будь-яким твердженням, що є істинним для даного поняття. У той час, як поняття лише вказують на предмет, про який іде мова в тексті, тези є змістовим або описовим наповненням бази знань. Від повноти наборів таких тверджень залежить повнота бази знань, а отже, і спроможність навчальної системи будувати ефективні завдання тесту. Фактично теза являє собою одне або кілька речень, в яких мова іде безпосередньо про деяке поняття, проте саме поняття тут словарно не фігурує. З точки зору лінгвістики теза, як правило, є реченням, з якого видалено підмет. Наведемо приклади: теза про поняття “процедура” “дозволяє розбити програму на підпрограми”; теза про поняття “клас” “може мати в своїй структурі не тільки поля-властивості, а й методи,

тобто функції і процедури” тощо. Множина тез у системі позначається такою формулою:

$$T = \{t_1, \dots, t_{n_2}\}.$$

Поняття і тези в сукупності будемо називати ПТ-елементами. Кожна теза стосується одного поняття. Цей зв’язок задається відношенням

$$CT: T \rightarrow C.$$

У свою чергу кожне поняття може мати довільну кількість тез, що описується відношенням

$$TC: C \rightarrow 2^T.$$

Як зазначалося вище, центральним носієм знань вважається навчальний матеріал. Тому виділення конкретних семантичних одиниць тісно пов’язане з процесом маніпулювання навчальним матеріалом. При підготовці методичних матеріалів дистанційного курсу весь матеріал врешті-решт прийнято ділити на дрібні фрагменти, які також називають кадрами. Таким чином, на кожному кроці видачі навчальної інформації студент отримує невеличкий фрагмент, що сприяє кращому зосередженню і сприйняттю поданого матеріалу. Поняття кадру є дуже важливим у ПТМ. З його допомогою забезпечується зв’язок семантичних елементів із навчальним матеріалом. У даній статті ми не зупиняємося на конкретній реалізації загальної структури збереження і подання навчального матеріалу, більше того, ПТМ допускає різні реалізації цієї структури, що дає можливість інтегрувати ПТМ із вже існуючими системами дистанційного навчання. Натомість, ПТМ разом з іншими моделями є однією із складових освітньої системи [14, 15], яка передбачає структурування навчального матеріалу за допомогою спеціальної моделі контенту [3–5]. Отже, позначимо множину фрагментів або сторінок навчального контенту такою формулою:

$$V = \{v_1, \dots, v_{n_3}\}.$$

Семантичні елементи ПТМ виділяються безпосередньо з тексту навчального фрагмента. Сам процес формування понятійно-тезисної бази фактично є осмисленим читанням навчального тексту разом із нескладними маніпуляціями щодо нього. Так, досліджуючи навчальний матеріал, викладач виділяє безпосередньо з тексту і додає в БЗ важливі навчальні поняття та їх тези. У результаті кожен фрагмент  $v_i$

може стати джерелом довільної кількості тез  $t_j$ , що задається відношенням

$$TV: V \rightarrow 2^T.$$

Кожна теза  $t_j$ , у свою чергу, стосується одного навчального фрагмента  $v_i$ :

$$VT: T \rightarrow V.$$

Оскільки тези стосуються лише одного навчального фрагмента, з якого вони були добуті, тоді як поняття можуть відноситися до багатьох навчальних ділянок, то зв'язок між поняттями і навчальним матеріалом забезпечується опосередковано через тези: поняття—теза—навчальний матеріал. Поняття, які стосуються даної навчальної ділянки, визначаються оператором:

$$CV(v) = \{c : TV(v) \cap TC(c) \neq \emptyset\}.$$

Відповідно, навчальний матеріал, якого стосується дане поняття, теж визначається оператором:

$$VC(c) = \{v : TV(v) \cap TC(c) \neq \emptyset\}.$$

Схематичне зображення ділянки ПТМ у співвідношенні з навчальним матеріалом показано на рис. 1.

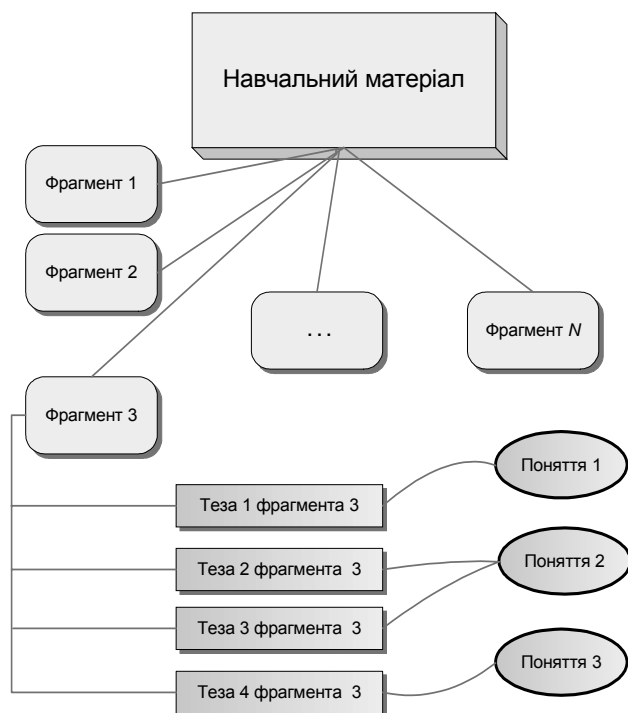


Рис. 1. Ділянка ПТМ у співвідношенні з навчальним матеріалом

## Класифікація тез і понять

Поняття і тези можуть бути віднесені до певних класів. Ця класифікація слугує для збереження в БЗ інформації про змістовний або лексичний характер того чи іншого поняття або тези. Класи понять і тез використовуються в алгоритмах побудови тестових завдань. Набір класів тез описується множиною

$$TClasses = \{tDefinition, tDestination, tEssence, tSyntax, tGeneral, tList, tCode, tAttaching\}.$$

ПТМ передбачає можливість розширення множини  $TClasses$  користувацькими класами тез, що дозволяє розширювати модель і налаштовувати її для різних предметних областей. Дано опис наявних класів із  $TClasses$ :

- *tDefinition* позначає тезу-визначення;
- *tDestination* позначає тезу-призначення; теза-призначення містить інформацію про призначення або роль того чи іншого поняття;
- *tEssence* описує так звану тезу-сутність, що слугує для передачі деякої принципової характеристики поняття;
- *tSyntax* позначає тезу-синтаксис, характерну для предметної області програмування; теза-синтаксис містить синтаксис деякої конструкції або команди мови програмування;
- *tGeneral* позначає загальну тезу, що подає відомість загального характеру про деяке поняття. Цей клас слід застосовувати у випадку, коли тезу не можна віднести до інших класів;
- *tList* позначає тезу-список, що слугує для виділення із тексту деякого списку об'єктів або компонент, які подають зміст або склад відповідного поняття;
- *tCode* позначає тезу, яка подає опис програмного коду, що міститься у відповідному понятті;
- *tAttaching* позначає службову тезу-прикріплення, яка вказує на те, що про відповідне поняття йде мова у фрагменті тексту.

Віднесення тези до певного класу відбувається за формулою

$$TClass = T \rightarrow TClasses.$$

Поняття, так як і тези, класифікуються з метою віднесення їх до певної змістовної групи. Множина класів понять визначається відношенням

$$CClasses = \{cGeneral, cCode\}.$$

Базовий набір класів містить два класи: *cGeneral* — для позначення поняття загального типу; *cCode* — для позначення поняття, яке подає програмний код. ПТМ передбачає можливість додавання класів для налаштування подання різних предметних областей. Поняття класифікуються за допомогою відношення

$$CClass = C \rightarrow CClasses.$$

### Структура тесту і тестових завдань

**Тест і його структура.** В основі генерації тестів у ПТМ лежить інформація про поняття, тези, навчальний матеріал та їх взаємозв'язок. Множина тестів задається так:

$$Test = \{test_i\}.$$

Вхідним параметром тесту є контрольна область навчального контенту, тобто набір фрагментів навчального матеріалу, за якими відбуватиметься тестування. Контрольна область контенту задається відношенням множини тестів і множини навчальних фрагментів:

$$Vtrg : Test \rightarrow 2^V.$$

Перш ніж розпочнеться побудова тестових завдань, здійснюється визначення так званої ресурсної області контенту, що слугуватиме додатковим джерелом ПТ-елементів, що можуть використовуватися як альтернативні відповіді для тестових завдань. Ресурсна область контенту задається відношенням

$$Vres : Test \rightarrow 2^V.$$

У простішому випадку ресурсна область контенту  $Vres(test)$  збігається з контрольною областю  $Vtrg(test)$ . Однак, у зв'язку з тим, що будівельним матеріалом тестових завдань є ПТ-елементи, можливі випадки, коли семантичних одиниць ПТМ буде недостатньо для побудови необхідної кількості завдань тесту. Крім того, використання ПТ-елементів з інших джерел може покращити якість тестових завдань. У таких випадках ресурсна область контенту  $Vres(test)$  визначається на основі пошуку навчальних фрагментів, матеріал яких відноситься до тієї ж предметної області, що й контрольний матеріал  $Vtrg(test)$ . Подібні алгоритми залежать від структури і можливостей моделі навчального контенту. Визначення предметної області в ієрархічно-мережевій моделі освітнього

контенту Tree-Net, з якою інтегрується ПТМ, розглянуто в попередніх працях автора [4].

**Тестове завдання.** ПТМ дає змогу генерувати завдання тесту на основі інформації про поняття, тези і їх зв'язок з навчальним матеріалом. Подамо опис технології автоматизованого створення тестових завдань. Множину тестових завдань позначимо у вигляді

$$Task = \{task_i\}.$$

Зв'язок тестів із завданнями задається відношенням

$$TestTasks: Test \rightarrow 2^{Task}.$$

У свою чергу кожне завдання пов'язується з контрольною ПТ-парою, тобто таким поняттям і його тезою, які ляжуть в основу цього тестового завдання. Зв'язок завдання з контрольною ПТ-парою задається так:

$$TaskCT: Task \rightarrow CT.$$

Існує набір шаблонів для тестових завдань  $TTempl = \{TTempl_i\}$ , які, крім того, можуть розширюватись користувачем, що дає змогу доповнювати різні предметні області новими типами тестових завдань. Шаблони, які застосовуються для тестових завдань, задаються в такому вигляді:

$$TaskTempl: Task \rightarrow TTempl.$$

Кожне завдання пов'язується з набором семантичних елементів, які використовуються як варіанти альтернативних відповідей. Залежно від шаблону як альтернативні відповіді можуть виступати або тези, або поняття. Зв'язок між завданнями і альтернативними варіантами задається відношенням

$$TaskAItems: Task \rightarrow 2^C \cup 2^T.$$

**Шаблон тестового завдання.** Шаблони призначені описувати спосіб побудови тестових завдань на основі ПТ-елементів. Можливість додавати нові шаблони дає змогу удосконалювати систему не тільки на етапі проектування, але й на етапі її використання для налаштування алгоритмів генерації тестів для різних предметних областей. Подамо структурні зв'язки і особливості використання шаблонів для побудови тестів на основі ПТМ.

Шаблони завдань поділяються на два типи: 1) в основі запитання лежить поняття, а в основі варіантів відповідей — тези; 2) в основі запитання лежить теза, а в основі варіантів від-

повідей – поняття. Таким чином, типізація запитального ПТ-елемента задається відношенням

$$TQEntity: TTempl \rightarrow CTEntity,$$

де  $CTEntity = \{Concept, Thesis\}$ .

Для задання шаблонного тексту, який відповідає за запитання тестового завдання, задається відношення шаблонів і відповідних елементів-зразків запитань:

$$TQStr: TTempl \rightarrow QStr,$$

де  $QStr = \{qStr_i\}$  – множина текстових шаблонів запитань.

Місце в тексті  $qStr_i$ , куди має бути вставлений текст запитального ПТ-елемента, позначається спеціальним чином, а саме символами “###”. Для позначення класів ПТ-елементів, які можуть бути застосовані для тестового завдання кожного з шаблонів, слугує відношення

$$TQClasses: TTempl \rightarrow 2^{CClasses} \cup 2^{TClasses}.$$

Як альтернатива  $TQClasses$  може бути застосоване зазначення класів, які заборонено використовувати в завданні, що означатиме дозвіл на використання всіх інших класів ПТ-елементів. Для цього слугує таке відношення:

$$TQNotClasses: TTempl \rightarrow 2^{CClasses} \cup 2^{TClasses}.$$

Для аналогічного зазначення класів для ПТ-елементів, що слугують як варіанти відповіді, використовуються відношення  $TAClasses$  і  $TANotClasses$ :

$$TAClasses: TTempl \rightarrow 2^{CClasses} \cup 2^{TClasses},$$

$$TANotClasses: TTempl \rightarrow 2^{CClasses} \cup 2^{TClasses}.$$

### Генерація тестового завдання

Подамо послідовність побудови і візуалізації тестового завдання  $task_k$  тесту на основі ПТ-елементів, їх зв'язку з навчальними матеріалами і шаблонів тестових завдань. Позначимо як  $test_k$  тест, в рамках якого будується завдання:

$$task_k \in TestTasks(test_k).$$

1. Вибір контрольної ПТ-пари. Випадковим чином вибирається контрольна ПТ-пара з контрольної області навчального контенту  $TaskCT(task_k) = (t_k, c_k)$ :

$$TaskCT(task_k) = (t_k, c_k) : (t_k, c_k) \in CT \wedge VT(t_k) \in Vtrg(test_k).$$

2. Пошук допустимих шаблонів завдань. На основі класів контрольного поняття  $c_k$  і тези  $t_k$  знаходиться множина всіх можливих варіантів шаблонів  $TT$ . По-перше, визначаються всі можливі шаблони для випадку, коли в основі тестового запитання лежить поняття:  $TQEntity(tt) = Concept$ . Позначимо цю множину шаблонів завдань  $TT'$  ( $TT' \subseteq TTempl$ ):

$$TT' = \{tt : TQEntity(tt) = Concept \wedge (CClass(c_k) \in TQClasses(tt) \vee TQClasses(tt) = 0) \wedge CClass(c_k) \notin TQNotClasses(tt) \wedge (TClass(t_k) \in TAClasses(tt) \vee TAClasses(tt) = 0) \wedge TClass(t_k) \notin TANotClasses(tt)\}.$$

Аналогічно знаходяться всі можливі шаблони для випадку, коли в основі тестового запитання лежить теза:  $TQEntity(tt) = Thesis$ . Цю множину шаблонів завдань позначимо  $TT''$  ( $TT'' \subseteq TTempl$ ):

$$TT'' = \{tt : TQEntity(tt) = Thesis \wedge (TClass(t_k) \in TQClasses(tt) \vee TQClasses(tt) = 0) \wedge TClass(t_k) \notin TQNotClasses(tt) \wedge (CClass(c_k) \in TAClasses(tt) \vee TAClasses(tt) = 0) \wedge CClass(c_k) \notin TANotClasses(tt)\}.$$

Таким чином, сукупна множина всіх можливих шаблонів завдань для завдання з контрольною ПТ-парою  $TaskCT(task_k)$  поєднає обидва випадки:

$$TT = TT' \cup TT''.$$

3. Вибір шаблону завдання. З множини шаблонів  $TT$  випадковим чином вибирається один шаблон  $tt_k$ , що й слугуватиме шаблоном завдання:  $TaskTempl(task_k) = tt_k$  для генерації і візуалізації тестового завдання  $task_k$ .

4. Пошук альтернативних варіантів відповідей. Відповідно до вибраного шаблону здійснюється пошук ПТ-елементів, що можуть слугувати як альтернативні варіанти відповіді. У випадку, коли  $TQEntity(tt_k) = Concept$ , можливі альтернативні варіанти відповіді вибираються з множини тез, клас яких відповідає параметрам шаблону  $tt_k = TaskTempl(task_k)$ :

$$TaskAItems' = \{t : t \neq t_k \wedge CT(t) \neq c_k \wedge (TClass(t) \in$$

$$\begin{aligned} &\in TAClasses(tt_k) \wedge TAClasses(tt_k) \neq 0) \vee \\ &\vee (t \in T \wedge TAClasses(tt_k) = 0) \wedge TClass(t) \notin \\ &\notin TANotClasses(tt_k) \wedge VT(t) \in Vres(test_k)\}. \end{aligned}$$

У випадку, коли  $TQEntity(tt_k) = Thesis$ , можливі альтернативні варіанти відповіді вибираються з множини понять, клас яких задовольняє вимоги шаблону  $tt_k = TaskTempl(task_k)$ :

$$\begin{aligned} TaskAItems' = \{c : c \neq c_k \wedge CT(t) \neq c_k \wedge ((CClass(c) \in \\ \in TAClasses(tt_k) \wedge TAClasses(tt_k) \neq 0) \vee \\ \vee (c \in C \wedge TAClasses(tt_k) = 0) \wedge CClass(c) \notin \\ \notin TANotClasses(tt_k) \wedge VC(c) \cap Vres(test_k) \neq 0)\}. \end{aligned}$$

З отриманої множини випадковим чином вибирається підмножина  $TaskAItems(task_k)$ , потужність якої залежить від кількості варіантів відповідей, що повинні ввійти в завдання.

5. Візуалізація тестового завдання. Коли всі семантичні дані готові, здійснюється подання тестового завдання користувачу.

Запитання будується на основі текстового шаблону  $TQStr(task_k)$ . У передбачене шаблоном місце вставляється текст запитального елемента контрольної ПТ-пари  $TaskCT(task_k) = (c_k, t_k)$ . У випадку, коли в основі запитання лежить поняття, тобто  $TQEntity(tt_k) = Concept$ , у запитанні фігуруватиме текст поняття  $c_k$ , в іншому випадку – текст тези  $t_k$ .

Список варіантів відповідей формується з набору неправильних відповідей  $TaskAItems(task_k)$  і правильної відповіді, яка отримується з контрольної ПТ-пари. Список варіантів відповідей сортується випадковим чином і разом із запитанням подається користувачу як тестове завдання. Процес генерації тестового завдання показано на рис. 2.

Після отримання відповіді студента на сформоване завдання відбувається аналіз правильності її порівнянням відповіді з істинним варіантом. У випадку неправильної відповіді дані про контрольні поняття  $c_k$ , тезу  $t_k$  і ділянку навчального матеріалу  $VT(t_k)$  використовуються для інформування студента про прогалини у вивченні відповідного навчального матеріалу і навчального поняття, а також для корекції подальшого сценарію навчання [16].

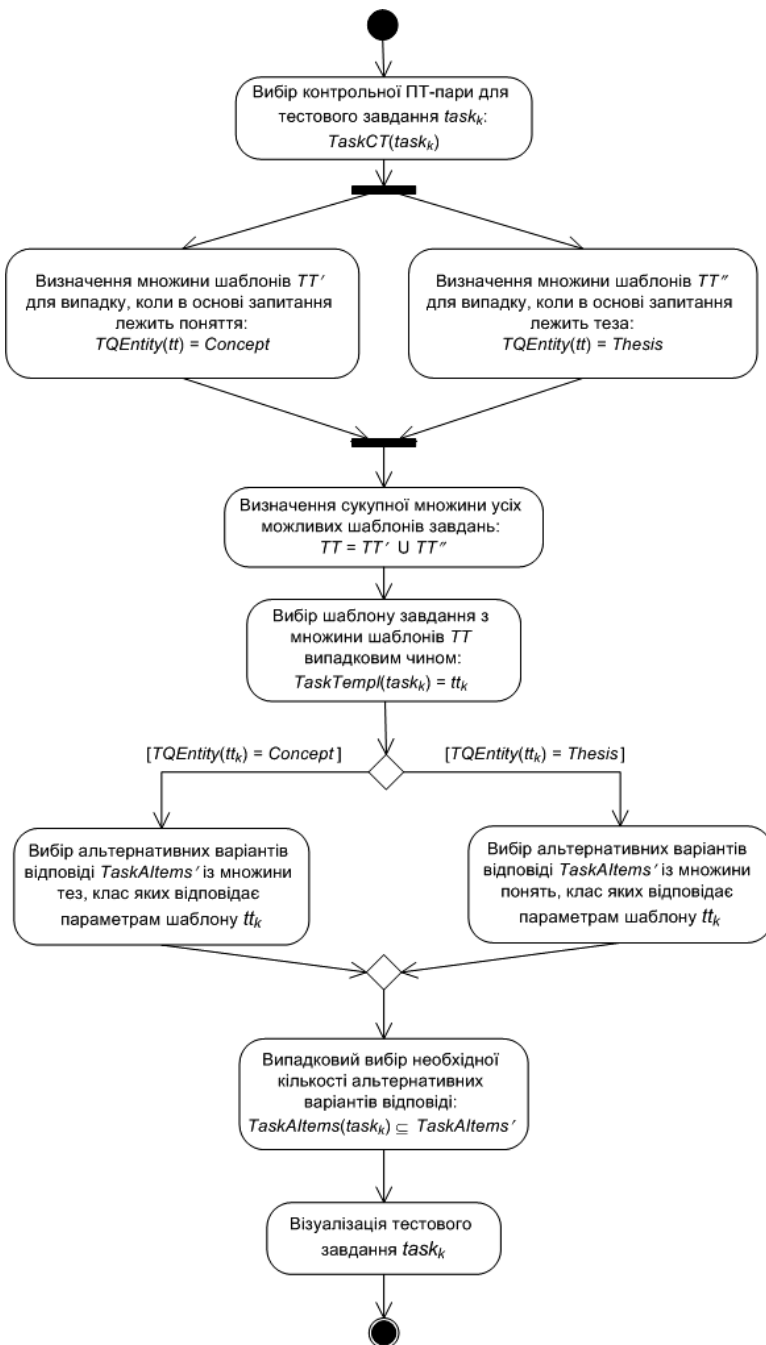


Рис. 2. Генерація тестового завдання. Діаграма діяльності в нотатції UML

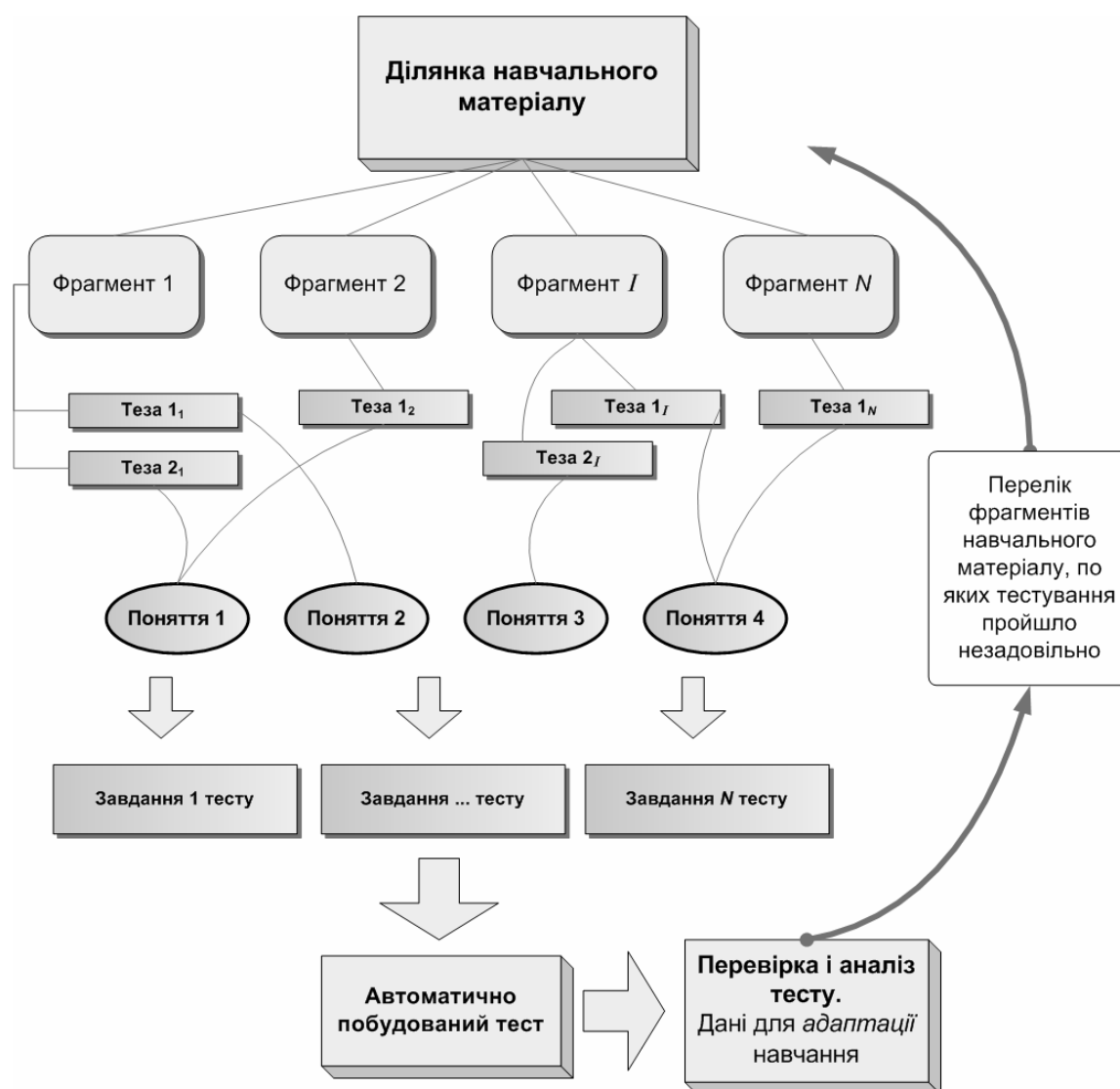


Рис. 3. Схема використання понятійно-тезисного підходу для автоматизації тестування

Таким чином, після проходження тесту з набору автоматично сформованих тестових завдань системою накопичується інформація про здобутки студента і його прогалини в навчанні. Виявлені прогалини можуть стати сигналом для детальнішої перевірки знань по відповідній ділянці навчального матеріалу або ж ступеня засвоєння відповідного навчального поняття. Все це дає змогу організовувати тестування, адаптивне відносно навчальної теми, поняття або розділу. Схема використання понятійно-тезисного підходу для автоматизації тестування зображена на рис. 3.

Вище було подано структуру і методи побудови тестового завдання множинного вибору.

ПТМ також дає можливість будувати тестові завдання інших типів, що розглянуто в [16–18].

### Порівняння ПТМ з іншими підходами побудови тестових завдань

Описані підходи до побудови тестових завдань були реалізовані і апробовані у відкритому навчальному процесі на порталі дистанційного навчання [19]. Практика використання ПТМ дає змогу зробити порівняльний аналіз технології ПТМ з традиційною технологією ручної побудови тестів та технологією автоматизованої побудови тестових завдань на основі семантичних мереж. Порівняльний аналіз наводиться в таблиці.



Таблиця. Порівняльний аналіз технологій побудови тестових завдань

Критерій	Технології тестування		
	Традиційний підхід	Семантичні мережі	Понятійно-тезисна модель
Принцип роботи	На основі педагогічної технології. Комп'ютерна реалізація традиційного тестування	На основі бази знань семантичної мережі. Центральною структурною ідеєю для формування тестових запитань є тріада: “поняття” – “відношення” – “поняття”. Технологія в рамках традиційної непедагогічної постановки задачі для ШІ	На основі понятійно-тезисної бази. Центральною структурною ідеєю для формування тестових завдань є зв'язок “навчальний матеріал” – “тези” – “поняття”. Модель носить педагогічний характер з елементами застосування інженерії знань
Спосіб формування бази знань або банку завдань	База формується ручним способом. Фактичне створення запитань і варіантів відповідей викладачем	Формування бази знань експертом з предметної області. Визначення понять і зв'язків між ними	Формування бази знань шляхом виділення із тексту навчального матеріалу семантичних одиниць за нескладною схемою
Трудові і часові витрати, необхідні для підготовки тесту	Витрати високі, прямо пропорційні кількості необхідних запитань. Важко оптимізуються	Необхідні великі витрати на формування бази знань. Проте формування тестів відбувається автоматично	Порівняно невеликі витрати. Генерація тестів відбувається автоматично
Необхідність залучення експертів для формування тестів або БЗ	У залученні експертів немає потреби. Формування тестів може здійснювати викладач відповідної дисципліни	Крім експерта з даної предметної області, потрібен інженер по знаннях	У залученні експертів немає потреби. Процес формування тестів легкий для сприйняття, його виконання може здійснювати викладач відповідної дисципліни
Якість тестових завдань і їх лексична зрозумілість	Якість висока, оскільки формування завдань здійснюється безпосередньо людиною	Завдання часто важкі для сприйняття людиною. Це пов'язано з тим, що семантичні мережі створювались для “пояснення” знань комп'ютеру, а тому погано підходять для спілкування з людиною	Хоча якість і поступається завданням, створеним вручну, проте вона достатньо висока, оскільки запитання будуються на основі фрагментів навчального тексту, створеного людиною
Ступінь автоматизації генерації завдань тесту	Автоматизація фактично відсутня. Використовуються готові завдання	Повна автоматизація на основі створеної семантичної мережі	Повна автоматизація на основі семантичних даних ПТМ і їх зв'язків з навчальним матеріалом
Відповідність тесту навчальному матеріалу закладається на етапі створення тестових завдань і залежить від конкретної системи.	Відповідність тесту навчальному матеріалу закладається на етапі створення тестових завдань і залежить від конкретної системи. У випадках потреби проводити тестування по деяких конкретних ділянках навчального курсу або його реструктуруванні з'являється необхідність збагачення тестових завдань мета-даними про навчальну ділянку	У зв'язку з тим, що для створення семантичної мережі застосовуються знання про предметну область взагалі, зв'язок з конкретними навчальними матеріалами має бути встановлений окремо. При цьому ускладнюється контроль відповідності тестового завдання конкретній ділянці навчального матеріалу	Зв'язок навчального матеріалу і семантичних даних закладено в самій моделі, і він задається природним чином під час формування бази знань. Це дає змогу при переструктуруванні навчального матеріалу коректно генерувати відповідні тестові завдання
Можливість вироблення рекомендацій для управління подальшим процесом навчання	Рекомендації виробляються залежно від ступеня інтеграції з навчальним матеріалом і наявності в тестових завданнях мета-даних про навчальну ділянку	Рекомендації можуть звертати увагу студента на конкретні поняття, які були слабо засвоєні ним під час вивчення	Завдяки глибокій інтеграції з навчальним матеріалом є можливість точно визначати, які ділянки навчального матеріалу потребують повторення, а також звернути увагу студента на конкретні навчальні поняття

## Висновки

Понятійно-тезисна модель дає можливість вирішити проблему несумісності задач традиційного штучного інтелекту і систем інтелектуального навчання через формалізацію знань з позицій обробки безпосереднього навчального тексту, який виступає первинним джерелом знань. Формування ПТ-бази відбувається завдяки осмисленому читанню навчального тексту, що значно спрощує процес побудови бази знань.

ПТМ забезпечує систему дистанційного навчання генератором тестових завдань, який у зв'язку з унікальністю кожного з побудованих тестів дає можливість звести до мінімуму проблему недоброчесного тестування, характерну для статичних тестів або тестів з недостатньо великим банком завдань. У той же час трудові витрати на побудову ПТ-бази менші за трудові витрати на побудову достатньо великого банку завдань і, відповідно, менші за витрати на побудову достатньо повної семантичної мережі предметної області. На відміну від тестових завдань, що генеруються на основі семантичних мереж, завдання на основі ПТМ мають кращу лексичну зрозумілість, що позитивно впливає на якість контролю. Точний зв'язок семантичних даних з ділянками навчального матеріалу дає змогу використовувати різноманітні стратегії при формуванні і оцінюванні тестів. Так, ПТМ дозволяє будувати тест по будь-якій вка-

заній ділянці навчального контенту, а також за результатами тестування, дає змогу точно визначити ті навчальні фрагменти і навчальні поняття, які потребують доопрацювання студентом. Такі можливості можуть бути використані для подальшої адаптації індивідуального навчального процесу на основі аналізу результатів тестування.

ПТМ інтегрується з ієрархічно-мережевою моделлю освітнього контенту Tree-Net, що дозволяє будувати індивідуалізоване навчаюче середовище з такими функціями, як генерація індивідуальних тестових завдань і генерація індивідуального навчального курсу [14, 15].

Практична реалізація ПТМ-підходів і автоматизоване тестування здійснені в рамках відкритого порталу дистанційного навчання [19]. Програмна реалізація містить засоби поповнення ПТ-бази на основі асинхронного Web-інтерфейсу і модуль генерації і аналізу тестів на основі ПТМ.

Подальші дослідження мають зосереджуватися на підвищенні якості тестових завдань і вдосконаленні методів і алгоритмів їх генерації. Перспективним тут є розширення базової класифікації тез і понять та формування на основі цієї класифікації нових шаблонів для завдань тесту. Ще одним перспективним напрямком є побудова моделі шаблону для тестового завдання відкритого типу. Поточні дослідження ПТМ висвітлюються на авторському сайті.

С.В. Титенко

ГЕНЕРАЦИЯ ТЕСТОВЫХ ЗАДАНИЙ В СИСТЕМЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ НА ОСНОВЕ МОДЕЛИ ФОРМАЛИЗАЦИИ ДИДАКТИЧЕСКОГО ТЕКСТА

Рассмотрен подход к автоматизации построения тестовых заданий на основе формализации учебного текста с помощью понятийно-тезисной модели (ПТМ). Представлены методы генерации тестовых заданий на базе ПТМ, рассмотрена структура модели, теста, тестового задания и его шаблона. Предложенный подход сравнивается с другими способами построения тестовых заданий.

S.V. Tytenko

TEST TASKS GENERATION IN THE DISTANCE LEARNING SYSTEM ON THE BASIS OF DIDACTIC TEXT FORMALIZATION MODEL

This paper suggests an approach to automation of test tasks generation based on the instructional text formalization by concept-thesis model (CTM). Presented are the methods of test tasks building on the basis of CTM. Also considered here is the structure of the model, test, task and tasks' templates. Furthermore, the proposed approach is compared to other techniques of test tasks building.

1. *Brusilovsky P. and Miller P.* Web-based testing for distance education // *Proceedings of WebNet'99, World Conference of the WWW and Internet, Honolulu, HI, Oct. 24–30, 1999, AACE / P. De Bra and J. Leggett (eds.).* – P. 149–154.
2. *Аванесов В.С.* Композиция тестовых заданий: Учебная книга для преподавателей вузов, учителей школ, аспирантов и студентов пед. вузов. — 2-е изд., испр. и доп. — М.: Адепт, 1998. — 218 с.

3. *Титенко С.В., Гагарін О.О.* Ієрархічно-мережева модель управління освітнім контентом системи безперервного навчання // Системний аналіз та інформаційні технології: Матер. X Міжнар. наук.-техн. конф. (20–24 травня 2008 р., Київ). – К.: НТУУ “КПІ”, 2008. – 258 с.
4. *Титенко С.В., Гагарін О.О.* Моделювання області знань в системі безперервного навчання на основі інтеграції моделі контенту Tree-Net і понятійно-тезисної моделі // VIII междунар. конф. “Интеллектуальный анализ информации ИАИ-2008”, Киев, 14–17 мая 2008 г.: Сб. тр. / Ред. кол.: С.В. Сирота (гл. ред.) и др. – К.: Просвіта, 2008. – С. 475–484.
5. *Титенко С.В., Гагарін О.О.* Формування навчального контенту на основі моделі даних Tree-Net // Матер. Всеукр. наук.-техн. конф. “Комп’ютерна математика в інженерії, науці та освіті” (CMSEE-2007), м. Полтава, 28–30 листопада 2007 р. – Полтава: Вид-во ПолНТУ, 2007. – 42 с.
6. *Pathak S., Brusilovsky P.* Assessing Student Programming Knowledge with Webbased Dynamic Parameterized Quizzes // ED-MEDIA’2002 – World Conference on Educational Multimedia, Hypermedia and Telecommunications, Denver, CO, June 24–29, 2002 / Barker P. and Rebelsky S. (eds.). – P. 1548–1553.
7. *QuizPACK* – Quizzes for Parameterized Assessment of C Knowledge. – <http://www.sis.pitt.edu/~taler/QuizPACK.html>
8. *Елизаренко Г.Н.* Проектирование компьютерных курсов обучения: концепция, язык, структура. – К.: НТУУ “КПИ”, 2001.
9. *Stankov S., Žitko B. and Grubišić A.* Ontology as a Foundation for Knowledge Evaluation in Intelligent E-learning Systems // AIED’05 Workshop SW-EL’05: Applications of Semantic Web Technologies for E-Learning. Papers of 12<sup>th</sup> International Conference on Artificial Intelligence in Education (AIED 2005). – Amsterdam, 2005. – <http://hcs.science.uva.nl/AIED2005/W3proc.pdf>
10. *Berners-Lee T.* Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. – The MIT Press, 2005.
11. *Гагарін О.О., Гайдаржи В.І., Титенко С.В.* Концептуальний підхід до подання знань в інтелектуальній освітній системі // Сучасні тенденції розвитку інформаційних технологій в науці, освіті та економіці: Матер. Всеукр. наук.-практич. конф., 11–13 грудня 2006 р., м. Луганськ. – Луганськ: Альма-матер, 2006. – С. 17–19.
12. *Артур К.* Как изучать Библию. – СПб., 1998.
13. *Gordon D. Fee, Douglas Stuart.* How to Read the Bible For All its Worth: A Guide to Understanding the Bible. – Michigan, 1982.
14. *Gagarin A., Tytenko S.* Complex model of educational hypermedia environment for ongoing learning // Образование и виртуальность-2007: Сб. науч. тр. 11-й Междунар. конф. Укр. ассоциации дистанционного образования / Под общ. ред. В.А. Гребенюка, Др. Киншука и В.В. Семенца. – Харьков–Ялта: УАДО, 2007. – С. 140–145.
15. *Гагарін О.О., Титенко С.В.* Проблеми створення гіпертекстового навчаючого середовища // Вісн. Східноукр. нац. ун-ту ім. Володимира Даля. – 2007. – Ч. 2, № 4(110). – С. 6–15.
16. *Гагарин А.А., Луценко А.Н., Титенко С.В.* Организация дистанционного обучения как информационный фактор реализации научно-технологической составляющей экономической безопасности государства // Экономическая безопасность государства и информационные технологии в ее обеспечении / Под общ. ред. Г.К. Вороновского, И.В. Недина. – К.: Знання України, 2005. – С. 608–619.
17. *Титенко С.В., Гагарін О.О.* Семантична модель знань для цілей організації контролю знань у навчальній системі // Сб. тр. междунар. конф. “Интеллектуальный анализ информации-2006”. – К.: Просвіта, 2006. – С. 298–307.
18. *Портал знань* – застосування ПТМ-підходів для автоматизації Web-тестування. – <http://www.znannya.org>
19. *Лабораторія СЕТ* – віртуальна лабораторія новітніх інформаційних технологій. Дослідження в області дистанційного навчання. – <http://www.setlab.net>